Gazi Üniv. Müh. Mim. Fak. Der.
Cilt 29, No 4, 677-687, 2014

Journal of the Faculty of Engineering and Architecture of Gazi University
Vol 29, No 4, 677-687, 2014

KÜRESEL EN İYİ YAPAY ARI KOLONİ ALGORİTMASI İLE OTOMATİK KÜMELEME

Celal ÖZTURK, Emrah HANCER, Dervis KARABOGA
Bilgisayar Müh. Bölümü, Mühendislik Fakültesi, Erciyes Üniversitesi, 38039, KAYSERİ
celal@erciyes.edu.tr, emrahhancer@erciyes.edu.tr, karaboga@erciyes.edu.tr

(Geliş/Received: 03.04.2013; Kabul/Accepted: 23.10.2014)

ÖZET

Kümeleme, verilerin analiz edilmesi için önemli bir teknik olup görüntü işleme ve istatistiksel veri analizi başta olmak üzere birçok alanda kullanılmaktadır. Özellikle son yıllarda kümeleme probleminin çözümüne yönelik olarak yapılan çalışmaların arttığı görülmektedir. Bu çalışmada, otomatik kümeleme problemini çözmek amacıyla yapay arı koloni algoritmasının küresel araştırma kabiliyeti geliştirilmiş ve algoritmanın vektörel araştırma yapabilmesi sağlanmıştır. Önerilen yöntem en çok bilinen data ve görüntü setleri üzerinde test edilmiştir. Alınan sonuçlar neticesinde önerilen metodun diğer metotlara oranla daha iyi bir performans sağladığı ve otomatik kümeleme problemlerinin çözümünde rahatlıkla kullanılabileceği görülmüştür.

Anahtar Kelimeler: Otomatik Kümeleme, Yapay Arı Kolonisi, Parçacık Sürü Optimizasyon

AUTOMATIC CLUSTERING WITH GLOBAL BEST ARTIFICIAL BEE COLONY ALGORITHM

ABSTRACT

Clustering, which is an important technique in analyzing data, is used in many fields, especially in image processing and statistical data analysis. In recent years, studies particularly on solving the clustering problem have been increased. In this paper, the global search ability of the artificial bee colony algorithm is improved and a vectorial search ability is integrated to the algorithm in order to solve the automatic clustering problem. The proposed clustering method is tested on the well-known benchmark datasets and images. The obtained results show that the performance of the proposed method is superior to the others and it can be applied to the automatic clustering problems.

Keywords: Automatic Clustering, Artificial Bee Colony, Particle Swarm Optimization

1. GİRİŞ (INTRODUCTION)

Son yıllardaki donanım ve yazılım alanlarında hatırı sayılır teknolojik gelişmeler sayesinde büyük boyutlu veriler kolayca saklanabilmektedir. Ancak bu verilerin çeşitli analiz ve değerlendirmelere tabi tutulabilmesi için bazı ön işlemlerden geçmesi gerekmektedir. Bu ön işlemler içinde kümeleme, verilerin değerlendirilmesi ve analizinde önemli bir yere sahiptir ve bilgisayar bilimleri, istatistik, tıp bilişimi ve görüntü işleme gibi birçok sahada geniş uygulama alanına sahiptir. Kümeleme bir veri setinin herhangi bir gözetim olmadan (Gözetimsiz Öğrenme) belirlenen kriterlere göre ayrılması işlemi olarak tanımlanan bir veri madenciliği disiplinidir. Bu kriterler veri seti elemanları arasındaki uzaklık mesafesi, benzerlik ve komşuluk gibi ölçütler temel alınarak tanımlanır. Kümeleme işleminden beklenen birbirine benzer veya yakın konumdaki elemanları aynı kümelerde toplaması ve birbirinden farklı veya uzak konumdaki elemanları ise farklı kümelerde toplaması gerekmektedir.

Kümeleme metotları temel olarak hiyerarşik (hierarchical) ve bölünmeli (partitioning) olmak üzere iki kategoride incelenmektedir. Hiyerarşik yapı bir kümede bulunan her bir bireyin bir sonraki kümede de yer alacağı prensibiyle oluşmaktadır. En üst seviyedeki kümede tüm bireyler yer alırken alt seviyelerdeki kümelerde daha az birey yer almaktadır.

En alt seviyede bulunan kümelerde ise sadece bir birey yer almaktadır [1]. Ara katmanda bulanan bütün kümeler alt ve üst katmandaki kümelerin birleşmesi veya bölünmesiyle oluşmuştur. Sonuç olarak hiyerarşik kümeleme metotları, bireylerin iç içe sınıflandırma ilişkisini ve sınıflandırmaların benzerlik seviyelerindeki farklılığı ağaç yapısı ile gösteren dendrogram oluşturma prensibiyle çalışmaktadır [2]. Hiyerarşik kümeleme metotları dendrogram oluşturma prensibine göre birleştirici (agglomerative) ve ayrıştırıcı (divisive) olmak üzere iki grupta incelenir [3]. Birleştirici metotlar, en alttaki katmandan üst katmana doğru kümelerin birleştirilmesine dayalı bir yol izler. Ayrıştırıcı metotlar ise, birleştirici metotların aksine en üstteki seviyeden alt seviyelere doğru kümelerin ayrıştırılması prensibine bağlı olarak çalışır. Hiyerarşik kümeleme metotları başlangıç koşullarından fazla etkilenmez ve bu metotlarda kümeleme sayısının bilinmesine gerek yoktur. Fakat bu metotların çalışma zamanı (runtime) performansı yavaştır ve bir sınıfa atanan birey dinamik olarak tekrar farklı bir sınıfa atanamamaktadır [4].

Bölünmeli kümeleme metotları ise n adet bireyden oluşan veri setini k (k<n) tane kümeye ayıran metotlardır [5]. Bölünmeli metotlar tanımlanan benzerlik kriterleri doğrultusunda bireyleri kümelere atarlar. Minkowski uzaklığı (Öklid mesafesi), Pearson ilişkisi, Mahalanobis uzaklığı ve Tchebyschev uzaklığı en çok kullanılan benzerlik kriterlerdir. Aynı kümenin elemanı olan bireyler arasındaki benzerlik maksimum ve farklı kümenin bireyleri arasındaki benzerlik minimum olacak şekilde kümeleme işlemi gerçekleştirilir [6]. Benzerliğin maksimize ve minimize edilmesi prensibi nedeniyle bölünmeli kümeleme, optimizasyon problemi olarak da ele alınabilir [7]. Bölünmeli kümeleme algoritmalarında kümeler, merkezler (centroid) vasıtasıyla ifade edilir ve birbirlerinden ayırt edilirler. Hiyerarşik kümeleme metotlarının dezavantajları bölünmeli kümeleme metotlarının avantajları, hiyerarşik kümeleme metotlarının avantajları ise bölünmeli kümeleme metotlarının dezavantajlarıdır. Ayrıca hiyerarşik kümeleme metotlarına göre daha çok kullanım alanına sahip olması nedeniyle bu çalışmada bölünmeli kümeleme metotları üzerine yoğunlaşılmıştır.

En çok bilinen bölünmeli kümeleme metotlarından birisi Macqueen tarafından önerilen K-means kümeleme metodudur [8]. K-means, k tane merkezden oluşan bir başlangıç çözümünü alır, bu merkezlere en yakın uzaklıktaki veri seti bireylerini bu kümelere atar ve kümelerin merkezleri tekrar bu bireylere bağlı olarak günceller. Küme merkezlerinde bir değişme olmayıncaya kadar veya durdurma kriteri sağlanıncaya kadar bu işlem tekrarlanır. K-means basit ve hızlı bir algoritma olmasına rağmen başlangıç koşullarına bağlı ve gürültüye duyarlıdır. Araştırmacılar K-means'ın dezavatajlarını gidermek amacıyla bu metot üzerinden birçok yeni metot geliştirmişlerdir. Bu metotların en çok bilinenlerinden biri Fuzzy C-Means'tir [9]. Fuzzy C-Means bulanık mantık teorisini temel alarak tasarlanmıştır. Bu kümeleme mantığı veri setindeki her birey için mevcut kümelere ait üyelik katsayısı tanımlanması ve bu üyelik katsayılarının güncellenmesine dayanmaktadır. Güncelleme işlemi sona erdiğinde bireyler hangi küme için daha fazla üyelik katsayısına sahipse o kümelere atanmaktadır. Fuzzy C-Means, K-means'tan daha iyi bir performans göstermesine rağmen başlangıç koşullarından etkilenmektedir ve gürültüye duyarlıdır. Bu algoritmaların dışında, K-medoids, CLARA ve CLARANS da popüler bölünmeli kümeleme metotları arasında yer almaktadır [10].

Fuzzy C-Means metodunun ortaya çıkmasıyla kümeleme metotları için yeni bir gruplama tanımlanmıştır: sert (crisp) ve yumuşak (fuzzy) kümeleme metotları. Sert metotlar bir bireyin yalnızca bir kümeye atanabilmesi, kümelerin boş küme olmaması ve kümelerin ortak elemana sahip olmaması kural setlerine dayalı olarak çalışırlar. Yumuşak metotlar ise sert metotların tam tersi işleve sahiptir. Bir birey mevcut bütün kümelerle her bir küme için sahip olduğu üyelik katsayıları ile alakadardır.

Bölümün başında bahsedildiği gibi yaşanan teknolojik gelişmeler sonucunda büyük boyutlu veriler rahatlıkla saklanabilmektedir. Bu veriler ayrıca dinamik özellik göstermekte olup sürekli güncellenebilmekte ve buna bağlı olarak da farklı özelliğe sahip bireylerin dahil olmasıyla küme sayısında da artma söz konusu olabilmektedir. Ayrıca günümüzde kullanılan verilerin birçoğunun küme sayısı da bilinememektedir. Dolayısıyla verilerin kümeleme işleminin, dinamik özellik göstermeyen hiyerarşik kümeleme veya kullanıcı tarafından küme sayısı isteyen bölünmeli kümeleme metotları ile yapılması mümkün değildir. Bir diğer söylemle optimum küme sayısını otomatik olarak bulurken aynı zamanda veri seti elemanlarını kümelere doğru şekilde dağıtan yöntemlere ihtiyaç artmıştır. Bu amaçla araştırmacılar evrimsel ve sürü zekası temelli algoritmalar kullanarak otomatik diğer adıyla küme sayısına ihtiyaç duymayan ve aynı zamanda dinamik özellik gösteren kümeleme metotları geliştirmeye çalışmışlardır. Omran ve arkadaşları [11], problemi ayrık bir optimizasyon problemi olarak ele almışlar ve parçacık sürü optimizasyon (PSO) algoritmasının ikili versiyonunu görüntü verileri üzerine uygulamışlardır. Merkezlerden oluşan büyük bir başlangıç kümesi oluşturulmuş ve merkezlerin seçimi ikili PSO'daki çözümlerin 0 ve 1 değerlerine göre yapılmıştır. İkili PSO işlemini tamamladıktan sonra K-means metodu çalıştırılarak başlangıç şartlarının etkisinin minimize edilmesi amaçlanmıştır. Elde edilen sonuçlar neticesinde PSO'nun UFA ve SNOB programlarından daha iyi sonuçlar verdiği görülmüştür. Das ve arkadaşları [12], diferansiyel gelişim algoritmasının

ölçekleme faktörünü (F) 0,5 ve 1 arasında rasgele sayılar üretilmesinin daha iyi olacağı sonucuna varmış ve diferansiyel gelişim algoritmasının bu formunu hem veri setleri hem de görüntü setleri üzerinde uygulayarak otomatik bir kümeleme metodu (ACDE) önermişlerdir. ACDE metodunda her çözüm 0-1 arasında rasgele üretilmiş sayılardan oluşur ve her çözümün karşılığında rasgele oluşturulmuş merkezler kümesi vardır. Merkezlerin seçimi aktivasyon temelli olarak yapılır. Yani 0,5 sayısından büyük olan merkezler işleme dahil edilirken, küçük olanlar dahil edilmez. Das ve arkadaşları [13], ACDE algoritmasını bir fuzzy kalite indeksi (validity index) ile uygulayarak da kümeleme işlemini gerçekleştirmiştir. Maulik ve Saha [14], diferansiyel gelişim algoritmasını uzaktan algılama resimlerinin otomatik kümelenmesi için kullanmışlardır. Ouadfel ve arkadaşları [15], PSO'nun mutasyon operatörleri ile geliştirilmiş bir versiyonunu (AMCPSO) görüntü verileri üzerinde uygulamışlar. Performans olarak AMCPSO algoritmasının GA'nın bir versiyonu olan VGA algoritmasına göre daha iyi sonuç verdiği görülmüştür. Kuo ve arkadaşları [16], PSO ile GA'dan oluşan bir hibrit model önermişler ve bunu istatistiksel veri setlerinin kümelenmesinde kullanmışlardır. Önerilen hibrit model ikili PSO ve GA algoritmalarının paralel çalışması ve elit seçme yöntemi ile popülasyon bireylerinin yenilenmesi üzerine kurulmuştur. Bu model ile PSO'nun erken yakınsama probleminin ortadan kaldırılması amaçlanmıştır. Algoritmanın çalışmasının bitmesinin ardından en iyi bireyin merkezlerine K-means uygulanarak başlangıç koşullarının negatif etkisi azaltılmaya çalışılmıştır. Bandyophadhyay ve arkadaşları [17] otomatik kümeleme problemini çok amaçlı bir optimizasyon problemi olarak ele almışlar ve buna yönelik olarak Xie-Beni indeksi (XBI) [18] ve Je (kuantalama hatası) senkronize olarak kullanılarak yeni bir amaç fonksiyonu önermişlerdir. Bu amaç fonksiyonunun minimizasyonu NSGA-II algoritması ile gerçekleştirilmiştir. Yine bir başka çalışmalarında XBI ve Sym [19] indeksleri senkronize bir şekilde kullanılarak AMOSA [20] algoritması ile otomatik kümeleme yaklaşımı önerilmiştir [21]. Bu yöntemlerden biraz farklı olarak Saha ve Bandyophadhyay [22] kümeleri alt kümelere ayırarak muhtemel çözümlerde temsil etmişlerdir. Bu alt kümeler ayrı ayrı ele alınarak veri seti elemanlarının atanması sağlanmış ve akabinde de uygun olan alt kümeler birleştirilerek işleme devam edilmiştir.

Yapay arı kolonisi (artificial bee colony (ABC)) algoritması nümerik problemlere [23], kablosuz sensör ağlarına [24], güç sistem denetleyicisi tasarımı [25], fuzzy veri kümeleme [26] ve görüntü analizi [27, 28] gibi alanlarda başarılı bir şekilde uygulanması, algoritmanın bu çalışma için seçilmesinde teşvik edici olmuştur. ABC algoritması 2005 yılında Karaboga [29] tarafından önerilmiş olup koloni halinde yaşayan arıların zeki davranışlarının modellenmesi sonucu ortaya çıkmıştır. Bu çalışmada, ABC algoritmasının üzerinde yapılan bazı değişikliklerle ABC'nin otomatik kümeleme işlemine entegre edilmesi sağlanmıştır. ABC algoritması üzerinde yapılan modifikasyon iki bölümden oluşmaktadır: çözümlerin vektörel değişimi (tüm parametrelerde değişim) ve küresel en iyi çözümün aktif olarak komşu çözüm bulmada kullanılması. Vektörel değişim ile yeni çözümlere daha hızlı erişim ve küresel en iyi çözüm özelliği ile de ABC algoritmasının en iyi çözüm bölgesinde araştırma özelliğinin geliştirilmesi sağlanmıştır. Geliştirilen Gbest-ABC algoritması hem görüntülere hem de data setlerine uygulanmış ve algoritmanın kabiliyeti ABC ve PSO algoritmaları ile birlikte analiz edilerek değerlendirilmiştir.

Makalenin organizasyonu şu şekildedir: Giriş Bölümünü takiben Bölüm 2'de sürü zekası, yapay arı koloni algoritması ve parçacık sürü optimizasyon algoritması ile ilgili ön bilgi verilmiştir. Bölüm 3'te yapay arı koloni algoritması üzerinde yapılan değişikliklerle birlikte önerilen yöntem açıklanmıştır ve Bölüm 4'te de önerilen yaklaşımın otomatik kümelemeye uygulanışı anlatılmıştır. Bölüm 5'te deneysel çalışma için gerekli olan veri setleri ve parametre ayarları açıklanmıştır. Bölüm 6'da uygulama sonuçları ve ilgili değerlendirmelere yer verilmiştir. Bölüm 7'de ise çalışmanın genel değerlendirmesi yapılarak makale sonlandırılmıştır.

2. ÖN BİLGİLER (BACKGROUND)

2.1. Sürü Zekası (Swarm Intelligence)

Sürü zekası, canlı bireylerin birbirleri ile ve çevreleri ile olan etkileşimleri sonucu ortaya çıkan davranışların araştırılması ve modellenmesi ile ilgilenen bir yapay zeka disiplinidir. Karıncaların gitmiş oldukları yollara feremon madde bırakarak diğer karıncalara yol hakkında bilgi vermeleri, kuş ve balık sürülerinin birlikte hareket ederek konumlarını değiştirmeleri, bal arılarının danslarla haberleşmeleri sürü zekası disiplinin yoğunlaştığı sosyal davranış modellerinden sadece birkaçıdır. Sürünün birbirleri ile ve çevreleriyle etkileşerek göstermiş olduğu davranışların sürü zekası kapsamında değerlendirilebilmesi için şu özelliklere sahip olması gerekmektedir:

a) İş bölümü (Division of Labour): Sürüye ait bireylerin kendi aralarında merkeziyetçi ve dayatmacı bir yaklaşım olmadan yeteneklerine uygun olarak iş bölümü yapabilmeleri gerekmektedir. İnsan türünde benlik ve ego gibi özellikler olması nedeniyle yönlendirici bir yaklaşım olmadan insan türünün iş bölümü yapabilmesi zordur. Fakat sürü zekası kapsamında değerlendirilen arıların davranış modeli incelendiğinde arıların bir bölümü yeteneklerine göre rasgele olarak yiyecek arar, bir bölümü yiyecek

kaynağında yoğunlaşır ve bir bölümü de yiyecek kaynakları hakkında bilgi verir.

b) Kendi başına organize olabilme (Self Organization): Kendi başına organize olabilme, bir sürüdeki bireylerin diğer bireylerle etkileşimden aldıkları bilgileri kullanarak kendi başlarına işlev görerek sürünün bütününü etkilemeleridir. Bir bireyin sürünün diğer bireyleri ile etkileşiminde temel komşuluk bilgileri kullanılır. Kendi başına organize olabilen bireylerin temel olarak pozitif geri besleme (positive feedback), negatif geri besleme (negative feedback), rasgelelik-salınım (fluctation) ve etkileşim (multiple interaction) yapabilme yetenekleri vardır. Pozitif geri besleme ile sürü içerisinde belli görevlerde bulunan bireylerin sürü amacına uygun işlemler gerçekleştirmesi durumunda o işlemi yapan birey sayısı artırılır. Bir nevi iyi olana teşvik söz konusudur. Karıncaların feremon madde miktarının fazla olduğu hatlara yönelme eğilimi pozitif geri beslemeye güzel bir örnektir. Negatif geri besleme ile ise pozitif geri besleme etkisinin dengelenmesi için kullanılır ve sürü bireyleri sonucu iyi olmayan işlemlerden uzaklaşmaya başlarlar. Rasgelelik-salınım, bireylerin sürü içindeki davranışları keskin sınırlar ile belirlenmemesi ve bireyin alacağı kararlarda bir tolerans bandının veya rasgelelik olması durumudur. Mesela karıncaların her zaman feremon maddesinin fazla olduğu hatlara yönelmemesi, bazen de içgüdüsel olarak farklı hatlara yönelmesi söz konusudur. Yani karıncaların hareketlerinde bir rasgelelik de söz konusudur. Ve son olarak da etkileşim ile sürüdeki bireyler birlikte hareket ederek ve birbirlerinden etkilenerek zeki davranışların ortaya çıkmasını sağlarlar. Etkileşim, kendi kendine organize olan bireylerin en önemli özelliğidir.

Sürü zekası temelli algoritmaların ilk örnekleri karıncaların yiyecek bulma davranışlarının modellenmesiyle ortaya çıkan karınca koloni [30] ve kuş ve balıkların sürü halinde hareket modelinin oluşturulmasıyla önerilen parçacık sürü optimizasyon (PSO) [31] algoritmalarıdır. İki algoritma da ortaya çıktıkları günden itibaren birçok alanda başarı ile uygulanmış ve kullanılmıştır. Ancak karınca koloni algoritmasının yapısının daha çok ayrık optimizasyon problemlerine uygun olması ve PSO algoritmasının yerel yakınsama problemleri olması nedeniyle daha güçlü sürü zekası temelli algoritmaların modellenmesine ihtiyaç duyulmuştur.

2.2. Yapay Arı Kolonisi Algoritması (Artificial Bee Colony Algorithm)

2000'li yılların başından itibaren özellikle zeki sürü davranışı gösteren bal arılarının hareketlerinin modellenmesi üzerine araştırmalar yoğunlaşmıştır. Yapılan çalışmalar neticesinde arılar algoritması (bees algorithm) [32], sanal arı algoritması (virtual bee algorithm) [33], arı sistemi (bee system) [34], arı koloni optimizasyon (bee colony optimization) [35] ve yapay arı koloni algoritması (artificial bee colony) [23] gibi modeller geliştirilmiştir. Önerilen modeller içinde en yaygın kullanılanlardan biri yapay arı kolonisi (ABC) algoritmasıdır [36].

ABC algoritması arıların yiyecek kaynağı arama davranışları modellenerek geliştirilmiştir. Bir arada yaşayan arıların sayıca çok fazla olmalarına rağmen aralarındaki kusursuz iş bölümü ve kendi kendine organize olabilme yetenekleri sayesinde kovandaki işlerde aksama veya kovan içinde karışıklık olmaz [37]. Belli bir düzen içerisinde kendilerine kaynak arayan bal arılarının davranışlarından esinlenilerek geliştirilen ABC algoritması çok parametreli nümerik fonksiyonların optimizasyonunda kullanılmış; genetik algoritma ve diferansiyel gelişim algoritmalarına karşı daha iyi performansa sahip olduğu görülmüştür [23].

ABC algoritmasında üç farklı tür arı vardır: kaşif arı, işçi arı ve gözcü arı. Her işçi arı bir yiyecek kaynağından sorumludur. İşçi arıların sayısı gözcü arıların sayısına eşittir. Kaynakların tükenmesi durumunda işçi arılar kaşif arılara dönüşmekte ve kaşif arılar yeni kaynak aramaya başlamaktadır. Dolayısıyla kaynakların tükenmesi sonucunda işçi arıların o kaynakları terketmesi negatif geri besleme ve kaşif arıların yeni kaynaklar araması ise rasgelelik-salınımı ifade etmektedir. Ortaya çıkacak kaşif arı sayısı "limit parametresi" ile belirlenmektedir.ABC algoritmasında bir yiyecek kaynağının pozisyonu bir çözüme denk gelmektedir. Yiyecek kaynağının polen veya nektar açısından zenginliği ise çözümün amaç fonksiyonu vasıtasıyla ifade edilmektedir. İşçi arılar görevli oldukları yiyecek kaynakların araştırır ve yiyecek kaynakları ile ilgili bilgileri gözcü arılarla paylaşırlar. Gözcü arılar ise işçi arılardan aldıkları bilgiler doğrultusunda zengin yiyecek kaynaklarına doğru daha fazla yönelme eğilimi gösterirler. Dolayısıyla gözcü arılar pozitif geri besleme özelliği gösterirler.

ABC algoritması ilk adımda yiyecek kaynakları ve bu yiyecek kaynaklarına uygun pozisyonların üretilmesi ile işleme başlar. Yiyecek kaynağının üretilmesi denklem (1) ile gerçekleştirilir;

$$x_{ij} = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right) \qquad (1)$$

Burada x_{ij} i. kaynağın j. pozisyonunu, $i=\{1,2,...,SN\}$, $j=\{1,2,...D\}$, SN popülasyon büyüklüğü ve D toplam pozisyon sayısını ifade etmektedir. x_j^{max} ve x_j^{min} pozisyonların alt-üst limitlerini belirtir ve $rand(0,1)$ 0 ile 1 arasında rasgele üretilmiş bir sayıdır.

Başlangıç yiyecek kaynaklarının üretilmesinin ardından işçi arılar tarafından ilgili oldukları yiyecek kaynaklarının araştırılma aşaması başlar. İşçi arılar yiyecek kaynakları hakkındaki bilgileri (pozisyon ve kalite bilgisi) hafızalarında tutarlar ve ilgili olduğu yiyecek kaynaklarının komşuluğunda denklem (2) ile

yeni yiyecek kaynakları belirlerler. Belirlenen yiyecek kaynağının kalitesi (amaç fonksiyon değeri) hafızasında tutmuş olduğu yiyecek kaynağından daha iyi ise yeni yiyecek kaynağı hafızaya alınır ve eski yiyecek kaynağı hafızadan silinir.

$$v_{ij} = x_{ij} + \theta_{ij}(\mathrm{x_{ij}} - \mathrm{x_{kj}}) \quad (2)$$

Burada x_{ij} i. kaynağın tek bir pozisyonunun (j. pozisyon rasgele seçilmiş) rasgele seçilmiş k. kaynağın pozisyonu ile etkileşimli (farka dayalı) olarak değiştirilerek v_i yeni kaynağın bulunmasını ifade eder. θ_{ij} ise -1 ve 1 arasında rasgele üretilmiş rasgele bir sayıdır.

İşçi arılar ilgili oldukları kaynaklarda araştırma işlemini tamamladıktan sonra elde ettikleri bilgileri gözcü arılar ile paylaşırlar. Gözcü arılar işçi arılardan aldıkları bilgiler kapsamında kaynakların nektar miktarlarına bağlı olarak denklem (3) ve (4) ile hesaplanan olasılık değerlerine göre kaynak seçme işlemini gerçekleştirirler.

$$p_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_i} \quad (3)$$

$$fitness_i = \begin{cases} \dfrac{1}{1+fit_i}, & fit_i \geq 0 \\ 1 + abs(fit_i), & fit_i < 0 \end{cases} \quad (4)$$

Burada fit_i i. kaynağın nektar miktarını (amaç fonksiyon değerini), $fitness_i$ ise i. kaynağın nektar miktarının uygunluk değerini ifade etmektedir. Bir kaynağın uygunluk değeri arttıkça o kaynağın seçilme olasılığının artması söz konusudur. Gözcü arılar araştırma yapılacak kaynakları belirledikten sonra işçi arılar gibi komşuluk mekanizmasını kullanarak denklem (2) ile yeni kaynak bulurlar. Bulunan yeni kaynağın nektar miktarı kontrol edilir. Eğer yeni kaynağın nektar miktarı fazla ise yeni kaynak hafızaya alınır ve eski kaynak hafızadan silinir. Kısacası işçi ve gözcü arılar aç gözlü yaklaşım (greedy selection) gösterirler.

ABC algoritmasında her çevrimde tüm işçi ve gözcü arılar arama islemlerini tamamladıktan sonra daha iyi kaynaklar bulunamadığı zaman ilgili kaynakların ceza puanı bir artırılır. Eğer bir aşamada herhangi bir kaynağa ait daha iyi bir kaynak bulunursa o kaynağın ceza puanı sıfırlanır. Ceza puanı ABC algoritması tarafından "limit" parametresi ile ifade edilen değeri açarsa o kaynak nektarı tükenmiş olarak addedilir ve kaynak işçi arısı tarafından terk edilir. Nektarı tükenen kaynağın işçi arısı kaşif arı gibi davranır ve denklem (1) ile tükenmiş kaynağın yerine yenisini bulur.

2.3. Parçacık Sürü Optimizasyon (Particle Swarm Optimization)

Parçacık sürü optimizasyon (PSO) algoritması 1990'ların ortalarında Kennedy ve Eberhart [31] tarafından önerilmiş olup zeki sürü davranışlarının modellenmesi sonucu ortaya çıkan sürü zekası temelli algoritmalarından biridir. Önerildiği günden itibaren biyoloji, tıp ve bilgisayar grafiklerine kadar kendine geniş kullanım alanı bulmuştur. PSO algoritması kuş, balık gibi zeki sürülerin sosyal davranışları ile bireysel davranışları birleştirilerek modellenmiştir. Bu algoritmada bir çözüm parçacık (particle) pozisyonları ve parçacıkların birim zamanda yer değiştirmesi ise hız (velocity) ile temsil edilir. Hız ve parçacık pozisyonları denklem (5) ve (6) ile güncellenir.

$$v_i(t+1) = w_{init} v_i(t) + c_1 r_1 \big(Lbest(t) - x_i(t)\big) + c_2 r_2 \big(Gbest(t) - x_i(t)\big) \quad (5)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (6)$$

Burada $x_i(t)$ t anındaki i. parçacığın pozisyon vektörü, $v_i(t)$ t anındaki i. parçacığın hız vektörü, $Lbest(t)$ t anına kadarki i. parçacığın en iyi pozisyon vektörü, $Gbest(t)$ t anına kadar en iyi parçacık pozisyonu, c_1-c_2 küresel-yerel pozisyon geliştirme denge sabitleri, r_1-r_2 0 ve 1 arasında rasgele üretilmiş sayıları, w_{init} hız vektörü için ağırlıklandırma katsayısını ve $x_i(t+1)$ ise t+1 anındaki i. parçacığın pozisyonunu ifade etmektedir. Parçacık pozisyonları vektörel olarak durdurma kriteri sağlanıncaya kadar güncelleme işlemine devam eder.

3. KÜRESEL EN İYİ ABC ALGORİTMASI (GLOBAL BEST ABC ALGORITHM)

Sezgisel yaklaşım gösteren tüm algoritmaların küresel ve yerel araştırma özellikleri vardır. Başarılı çözümler üretilebilmesi için çözüm uzayında küresel araştırma ve yerel araştırmanın dengeli bir şekilde yürütülmesi gerekmektedir. Çözüm uzayında yerel araştırmaya daha fazla ağırlık verildiği takdirde çözümlerin birbirine aşırı benzemesine ve erken yakınsama problemine, çözüm uzayında küresel araştırmaya ağırlık verildiği takdirde ise çözümlerin birbirinden aşırı uzaklaşmasına ve iyi çözümleri bulamama problemlerine yol açmaktadır. ABC algoritmasında işçi ve kaşif arılar küresel araştırma özelliğini, gözcü arılar ise yerel araştırma özelliğini sağlamaktadır. Ayrıca gözcü arıların iyi kaynakları olasılıklı olarak seçmesi yerel ve küresel araştırma dengesini sağlayan diğer bir faktördür.

Bu çalışma ile PSO'nun sahip olduğu bazı özellikler ABC algoritmasına entegre edilerek daha güçlü bir algoritmanın ortaya çıkarılması sağlanmıştır. ABC algoritmasının farklı çözümler üretebilme ve araştırma kabiliyetlerinin artırılmasına yönelik olarak PSO algoritmasından esinlenilerek bir modifikasyon

(Gbest-ABC) geliştirilmiştir. İlk olarak bir pozisyonda yapılan değişim yerine tüm pozisyonlarda değişim yapılması öngörülmüştür. İşçi arıların vektörel olarak yeni kaynaklar bulabilmesi denklem (2) yerine denklem (7) ile sağlanmıştır.

$$v_i = x_i + \theta_i.(\mathrm{x_i} - \mathrm{x_k}) \quad (7)$$

Burada θ_i 0 ve 1 arasında rasgele sayılardan oluşan bir vektördür.
İkinci olarak gözcü arıların en iyi kaynağa yönelmesi ve bu kaynak etrafında arama yapması sağlanmıştır. Gözcü arıların küresel en iyi çözümün etrafında komşu kaynak araştırma mekanizması denklem (2) yerine denklem (8) entegre edilerek gözcü arıların iyi kaynaklar etrafında araştırma yapma eğilimi daha etkili hale getirilmiştir;

$$v_i = x_i + \theta_i.(\mathrm{x_i} - \mathrm{Gbest}) \quad (8)$$

Küresel en iyi çözüm etrafında aramanın sadece gözcü arılar tarafından gerçekleştirilmesinin ve işçi arılar tarafından kullanılmamasındaki temel amaç algoritmanın küresel araştırma özelliğinin korunmak istenmesidir. Ayrıca gözcü arıların temel ABC algoritmasında iyi çözümler komşuluğunda arama yapmaya daha meyilli olduğu da göz önüne alındığında en iyi çözüm etrafında arama vazifesinin sadece gözcü arılara verilmesinin isabetli bir karar olduğu görülmektedir. Önerilen modifikasyon ile geliştirilen ABC modeli, bu özelliği ile literatürdeki diğer küresel çözüm tabanlı ABC algoritmalardan ayrılmaktadır.

4. GBEST-ABC İLE OTOMATİK KÜMELEME (AUTOMATIC CLUSTERING WITH GBEST-ABC)

Bölünmeli kümeleme işlemi benzerlik kriteri üzerinden gerçekleştirilir. Benzerlik kriteri değerinin aynı kümenin elemanları arasında minimum olması, farklı kümelerin elemanları arasında ise maksimum olması beklenir. En yaygın kullanılan benzerlik kriterlerinden birisi Öklid mesafesidir. Bir veri setindeki iki birey arasındaki Öklid mesafesi şu şekilde tanımlanır:

$$d(p_v, p_y) = \sqrt{\sum_{j=1}^{N_d} (p_{v,j} - p_{y,j})^2} = \left\| p_v - p_y \right\|_2 \quad (9)$$

Burada p_v ve p_y veri setindeki v. ve y. bireyleri, $p_{v,j}$ ve $p_{y,j}$ v. ve y. bireylerin j. özniteliğini (feature), N_d toplam öznitelik sayısını ve $d(p_v, p_y)$ ise v. ve y. bireyler arasındaki Öklid mesafesini ifade eder.

Öklid mesafesi basit olmasına rağmen büyük veri setlerini kümeleme noktasında istenilen performansı sağlayamamaktadır. Dolayısıyla araştırmacılar veri setindeki bireyler arasındaki benzerliği temel alarak istatistiksel ve matematiksel kümeleme kalite indeksleri (clustering validity indexes) geliştirmişlerdir. Kümeleme kalite indekslerinin öncelikli kullanım alanı, kümeleme metotlarının kümeleme performanslarını ölçmek ve karşılaştırmaktır. Diğer kullanım alanı ise, bir veri setindeki küme sayısının belirlenmesidir.

Bir kalite indeksinin sahip olması gereken iki temel şart vardır: Aynı sınıf içindeki bireylerin birbirleri ile maksimum düzeyde benzerlik göstermeleri (compactness) ve kümelerin birbirleri ile aralarındaki benzerliğin minimum düzeyde olması (separatness). En çok bilinen kalite indeksleri Dunn indeks (DI) [38], Calinski-Harabasz indeks [39], DB indeks [40], CS metrik [41] ve VI indekstir [42]. Turi [42], VI indeksinin DB ve Dunn indekslerinden daha iyi olduğunu öne sürmüştür. Bu nedenle bu çalışmada VI indeksi seçilmiş ve bu indeks Gbest-ABC algoritması ile kullanılarak otomatik kümeleme işlemi gerçekleştirilmiştir.

VI indeksi iki modülden oluşmaktadır:

a) Küme içi uzaklık (Intra-Cluster Separation): Kümenin elemanı olan bireylerin küme merkezine olan uzaklıklarının ortalamasını ifade etmektedir. Küme içindeki bireylerin birbirlerine uzaklıklarının yakın olması prensibinden yola çıkılarak küme içi uzaklık modülünün minimize edilmesi beklenir.

$$intra = \frac{1}{N_p} \sum_{k=1}^{K} \sum_{p \in C_k} \|p - m_k\|^2 \quad (10)$$

Burada N_p veri setindeki toplam birey sayısını, K toplam küme sayısını, m_k k. kümenin merkezini ifade etmektedir. Her bireyin ait olduğu kümenin merkezine olan Öklid mesafesinin kareleri toplanıp ortalaması alınarak küme içi uzaklık modülü hesaplanmış olur.

b) Kümeler arası uzaklık (Inter-Cluster Separation): Kümeler arası uzaklıkların minimum olanını temsil etmektedir. Kümeler arası uzaklığın maksimize edilerek kümelerin birbirinden ayrıştırılması amaçlanır.

$$inter = \min\{\|m_k - m_{kk}\|^2\}, k \neq kk \quad (11)$$

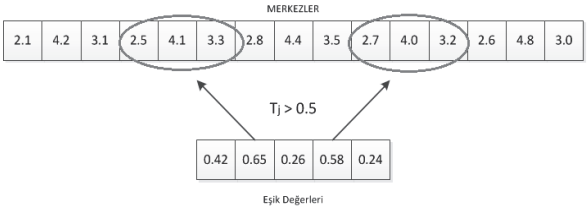
Bu iki modülün birleşimi ile VI indeksi ortaya çıkmıştır. VI indeksi denklem (12) ile ifade edilir.

$$VI = (c \times N(\mu, \sigma) + 1) \times \frac{intra}{inter} \quad (12)$$

Burada c kullanıcı tarafından tanımlı bir sabit ve $N(\mu, \sigma)$ K merkezi alarak μ ortalama, σ standart sapma ile hesaplanan normal dağılım değerini ifade etmektedir.

$$N(\mu,\sigma)=\frac{1}{\sqrt{2\Pi\sigma^2}}e^{[-\frac{(K-\mu)^2}{2\sigma^2}]} \quad (13)$$

Burada μ değeri 5 ve 6 kümeli verilerin kümelenmesi ile ilgilenildiği durumlarda 2, 5'ten az küme içeren verilerle ilgilenilmesi durumunda 1 veya 0 ve 5'ten fazla küme içerin verilerin kümelenmesi ile ilgilenilmesi durumunda ise 3 alınır [36]. σ ise genellikle 1 olarak seçilir.



Şekil 1. Merkez seçme mekanizması (The mechanism of centroid selection)

Gbest-ABC algoritmasının otomatik kümeleme problemine uygulamasında her bir kaynak çözüm veri setinden rasgele seçilmiş Kmax tane merkezden ve bu merkezlere ait 0-1 arasında rasgele üretilmiş eşik değerlerden oluşur. Herbir merkez noktasının d tane de özniteliği olduğu varsayılırsa toplam olarak her çözüm $K_{max} * d + K_{max}$ boyutunda ifade edilir. Kümelemede kullanılacak merkezlerin seçimi ise merkezlerin ilgili eşik değerlerinin değerlendirilmesi ile gerçekleşir. Şekil 1'de 3 özniteliği sahip 5 tane merkez noktası ve bu merkezlere ait 5 tane de rasgele üretilmiş eşik değerleri (T_j) vardır. Merkez seçim işlemi yine Şekil 1'de görüleceği gibi eşik değeri sadece 0,5'ten büyük olan merkezler alınarak gerçekleştirilmiştir. Eşik değerlere göre seçilen merkez sayısının 2'den az olması durumunda eşik değerler yeniden üretilir. Dolayısıyla VI indeksi hesaplanmasında oluşacak problemleri engellemek için bir veri setinin sahip olabileceği merkez sayısı en az 2 olarak belirlenmiştir (K_{min}=2). Bir veri setinin sahip olabileceği maksimum merkez sayısı ise 10 ile sınırlandırılmıştır (K_{max} =10). Merkez seçiminin

```
begin
    Populasyon sayisi, limit ve durdurma kriteri (MCN) belirle;
    Yiyecek kaynaklarini Denklem (1) ile uret;
    for cycle ← 1 to MCN do
        foreach isci-ari i do
            x_i kaynaginin merkezlerini (m_ik) T_j > 0.5 sarti ile Sekil 1'deki gibi sec;
            while Secilen merkez sayisi K < 2 do
                x_i kaynaginin esik degerlerini Denklem (1) ile yeniden uret;
                x_i kaynaginin merkezlerini (m_ik) T_j > 0.5 sarti ile Sekil 1'deki gibi sec;
            end
            for y = 1 to N_p do
                pattern p_y'nin butun kumelere olan Oklid mesafelerini (d(p_y, m_ik)) Denklem (9) ile hesapla;
                p_y'yi en yakin Oklid mesafesindeki kumeye ata;
            end
            VI(x_i) degerini Denklem (12) ile hesapla;
            x_i'nin komsulugunda rasgele secilmis x_k komsulugunda Denklem (7) ile v_i uret;
            x_i ve v_i arasinda ac gozlu yaklasim uygula;
        end
        foreach gozcu-ari i do
            Denklem (3) ve (4) ile p_i olasiliginda bir x_i yiyecek kaynagi sec;
            while Secilen merkez sayisi K < 2 do
                x_i kaynaginin esik degerlerini Denklem (1) ile yeniden uret;
                x_i kaynaginin merkezlerini (m_ik) T_j > 0.5 sarti ile Sekil 1'deki gibi sec;
            end
            for y = 1 to N_p do
                pattern p_y'nin butun kumelere olan Oklid mesafelerini (d(p_y, m_ik)) Denklem (9) ile hesapla;
                p_y'yi en yakin Oklid mesafesindeki kumeye ata;
            end
            VI(x_i) degerini Denklem (12) ile hesapla;
            x_i' ve gbest komsulugunda Denklem (8) ile v_i uret;
            x_i ve v_i arasinda ac gozlu yaklasim uygula;
        end
        if kaynakcezapuani > limit then
            Denklem (1) ile yeni kaynak uret.;
        end
        gbest kaynagini guncelle;
    end
end
```

Şekil 2. Gbest ABC algoritması ile otomatik kümeleme (Automatic clustering with the Gbest-ABC algorithm)

ardından veri setindeki bireyler kendilerine en yakın küme merkezlerine atanırlar ve kümelere atanan bireylerin VI indeksi denklem (12) ile hesaplanır. Hesaplanan VI indeksine göre eşik değerleri ve merkezler işçi arılar tarafından denklem (7) ile gözcü arılar tarafından denklem (8) ile açgözlü yaklaşım uygulanarak güncellenir. Merkez güncelleme işlemi, rasgele seçilmiş bir öznitelikte vektörel olarak gerçekleşir; çünkü tüm öznitelikler üzerinde gerçekleşen merkez güncelleme, merkezlerin yapısının bozulmasına neden olabilmektedir. Algoritmanın otomatik kümeleme problemine uygulamasının detaylı kaba kodu Şekil 2'de ayrıca sunulmuştur.

5. DENEYSEL DİZAYN (EXPERIMENTAL DESIGN)

Gbest-ABC algoritması 7 adet data seti ve 5 adet 8 bitlik griton resim olmak üzere toplam on iki veri setine uygulanmıştır. Önerilen metodun performans analizi ise kümeleme kalitesi ve optimal küme sayısı temel alınarak yapılmıştır. Gbest-ABC kümeleme metodunun uygulandığı data setleri sırasıyla Wisconsin (683 birey, 9 öznitelik, 2 sınıf), Iris (150 birey, 4 öznitelik, 3 sınıf), Wine (178 birey, 13 öznitelik, 3 sınıf), Vehicle (846 birey, 18 öznitelik, 4 sınıf), Dermatology (358 birey, 34 öznitelik, 6 sınıf) ve Glass (214 birey, 9 öznitelik, 6 sınıf) ve uygulandığı resimler Tahoe (300x300 piksel, 4 sınıf), MRI (300x300 piksel, 5 sınıf), Jet (512x512 piksel, 6 sınıf), Lena (512x512 piksel, 6 sınıf) ve Pepper (512x512 piksel, 7 sınıf) griton resimleridir. Tüm data setlerine kümeleme işleminden önce normalizasyon işlemi uygulanmıştır.

Önerilen yöntemin performans analizi temel ABC ve PSO algoritmalarının yanında, ekonomik emisyon dağıtım problemini (economic-emission dispatch problem) çözmek amacıyla küresel çözüm mekanizmasını kullanarak modifiye edilen ABC'nin bir başka versiyonu GABC [44] de deneysel çalışmaya dahil edilmiştir. GABC'nin dahil edilmesindeki temel amaç Gbest-ABC'nin küresel çözüm kullanmadaki etkinliğini daha net bir şekilde ortaya koymaktır.

Ele alınan algoritmaların parametre değerleri yapılan denemeler ve literatürdeki bilgiler doğrultusunda saptanmıştır. Gbest-ABC, GABC [44] ve ABC algoritmalarının kaynak sayısı 30 ve limit değeri 20 olarak belirlenmiştir. PSO algoritmasında ise parçacık sayısı 30, küresel ve yerel araştırma denge faktörleri c1 ve c2 1.49, hız ağırlıklandırması başlangıç değeri (wstart) 0.72 ve bitiş değeri (wend) 0.4 olarak belirlenmiştir. Görüntü kümeleme işlemlerinde Vmax eşik değerler için 1 ve merkezler için 255, data kümeleme işlemlerinde eşik değerler için yine 1 ve merkezler için ise data setlerinin normalize edilmesi nedeniyle 1 olarak alınmıştır. Tüm algoritmalar için iterasyon sayısı, görüntü kümelemede 500, data kümelemede ise 250 olarak belirlenmiştir.

6. DENEYSEL ÇALIŞMALAR (EXPERIMENTAL STUDIES)

Sonuçlar 30 sümulasyon üzerinden kalite indeksi ve optimal küme sayısı olarak ortalama ve standart sapma cinsinden Tablo 1'den Tablo 4'e kadar verilmiştir. En iyi sonuçlar kalın (bold) yazı tipinde ve standart sapmalar parantez içinde sunulmuştur. Ortalama optimal küme değerlerinin yanı sıra 30 koşmanın kaçında optimum küme sayısının elde edildiğine dair bilgi veren bar grafikleri de Şekil 3 ve 4'te verilmiştir.

6.1. Otomatik Görüntü Kümeleme (Automatic Image Clustering)

Görüntü kümeleme ortalama indeks ve optimal küme sonuçları sırasıyla Tablo 1 ve Tablo 2'de sunulmuştur. Tablo 1'de görüleceği üzere Gbest-ABC kümeleme kalitesi olarak diğer metotlardan Jet ve Pepper dışında daha iyi sonuç vermektedir. Jet ve Pepper resimlerinde ise en iyi ikinci performansa sahiptir. Diğer metotların elde ettikleri kümeleme kalitelerine baktığımızda, GABC Gbest-ABC'den sonraki ikinci en iyi performansı sağlamakta, PSO ve ABC de performans olarak onun ardından gelmektedir.

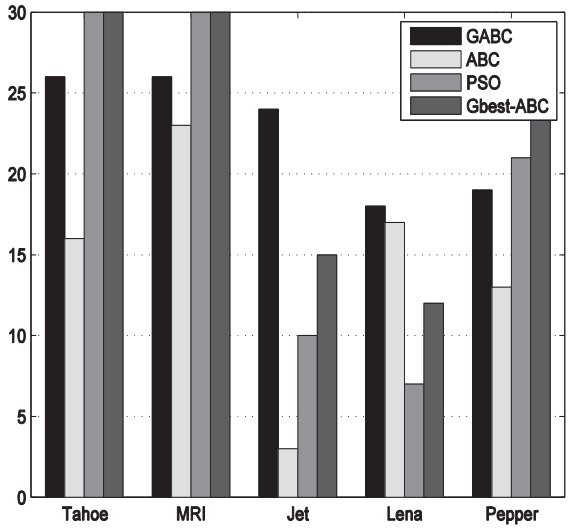
Tablo 1. Görüntü Kümeleme İçin VI İndeks Sonuçları (The Results of VI Index For Image Clustering)

Resimler	Gbest ABC	GABC	ABC	PSO
Tahoe	**0,0375** (7,21E-05)	0,0384 (0,0009)	0,0391 (0,0008)	**0,0375** (2,67E-06)
MRI	**0,0345** (0,0006)	0,0354 (0,0009)	0,0360 (0,0009)	0,0346 (0,0007)
Jet	0,0577 (0,0036)	0,0592 (0,0035)	**0,0543** (0,0014)	0,0591 (0,0032)
Lena	0,0784 (0,0069)	**0,0766** (0,0032)	0,0787 (0,0032)	0,0819 (0,0071)
Peppers	**0,0849** (0,0064)	0,0875 (0,0036)	0,1073 (0,0096)	0,08713 (0,0101)

Tablo 2. Görüntü Kümeleme İçin Ortalama Optimal Küme Sayıları (The Mean Numbers of Optimal Clusters for Image Clustering)

Resimler	No	Gbest ABC	GABC	ABC	PSO
Tahoe	4	**4** (0)	4,0333 (0,4901)	3,4667 (0,8995)	**4** (0)
MRI	5	**5** (0)	5,0333 (0,4901)	4,5667 (0,8584)	**5** (0)
Jet	6	5,6 (0,5085)	**5,8** (0,8050)	4,0667 (0,8277)	5,4 (0,5632)
Lena	6	**5,7** (0,5959)	6,4137 (0,5680)	5,6 (1,5887)	5,4 (0,6214)
Peppers	7	**7,0667** (0,4497)	6,9 (0,6047)	7,4333 (1,073)	7,2 (0,7611)

Elde edilen ortalama küme sayılarına baktığımızda ise, Gbest-ABC'nin bu konuda da başarısını sürdürdüğü Tablo 2'de açıkça görülmektedir. ABC'nin bir başka küresel en iyi çözümü kullanan versiyonu olan GABC ise sadece Jet'te daha iyi performans sağlamaktadır. Bunun yanında temel ABC ise ortalama küme sayısı açısından diğer metotlara oranla genel olarak kötü sonuçlar vermektedir. Hatta ABC Jet resmi için 4.06 gibi çok düşük bir ortalama küme sayısı bulmuştur. Bir diğer karşılaştırmada kullanılan PSO algoritması ise, Tahoe ve MRI resimlerinde Gbest-ABC ile beraber optimal küme sayısı bulmada en iyi performansı sağlarken, bu başarılı performansını diğer resimlere taşıyamadığı görülmektedir. Algoritmaların 30 koşmanın kaçında optimum küme sayısını elde ettiğine dair bar grafiği Şekil 3'te sunulmuştur. Tahoe ve MRI resimlerinde 30 da 30 bir oranla Gbest-ABC ve PSO'nun %100 ile yüksek başarı sağladığı, Jet ve Lena resimlerinde GABC'nin optimum küme sayısı elde etmede daha iyi olduğu ve Pepper resminde ise Gbest-ABC'nin daha fazla optimum küme sayısı bulduğu görülmektedir. Bu bilgiler ışığında Gbest-ABC'nin genel olarak diğerlerinden otomatik görüntü kümelemede daha iyi performans sağladığı görülmektedir.



Şekil 3. Resim kümeleme için 30 koşma üzerinden elde edilen toplam optimum küme sayısı (The total numbers of obtained optimum clusters for image clustering over 30 simulations)

6.2. Otomatik Data Kümeleme (Automatic Data Clustering)

Data kümeleme için ortalama kalite indeks ve optimal küme sonuçları Tablo 3 ve 4'te sunulmuştur. Algoritmalar data kümeleme performansı çerçevesinde değerlendirildiğinde de Gbest-ABC elde edilen kümeleme kalitesi olarak diğer algoritmaların hepsinden iyi olduğu Tablo 3'te gözlemlenmektedir. Diğer algoritmaların performanslarına bakıldığında ise, 2 kümeli data setlerinde PSO ABC ve GABC'ye göre daha iyi kümeleme kalitesi sağlarken, 2'den fazla küme içeren data setlerinde ABC ve GABC PSO'dan daha iyi kümeleme kalitesi sağlamaktadır.

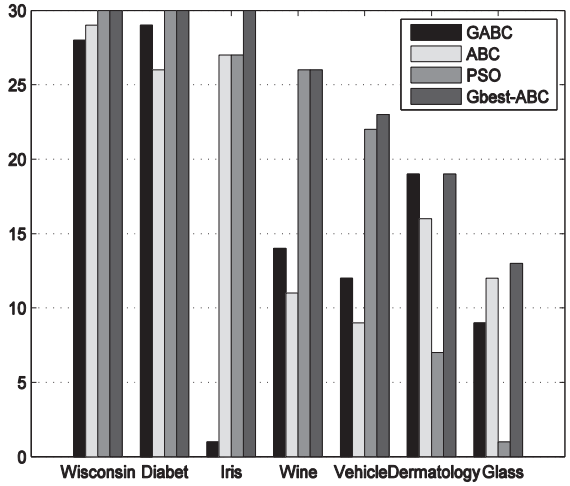
Elde edilen ortalama optimal küme sayılarına bakıldığında ise, Tablo 4'te görüldüğü gibi Gbest-ABC Dermatology ve Glass dışında en iyi performansı sağlamaktadır. Yine Tablo 4'ten görüleceği üzere PSO Wisconsin, Diabet ve Wine data setlerinde iyi performans gösterirken, daha büyük kümeli data setlerinde performansı düşmektedir. GABC ve ABC ise ortalama optimal küme sayısı elde etmede sadece Dermatology ve Glass data setlerinde en iyi performansı sağlarken, diğer data setlerinde genel olarak performansı düşüktür. GABC'nin Iris data setinde hiçbir zaman 3 küme sayısını bulamaması ve ABC'nin Vehicle veri setinin kümelemesinde 5.13 gibi bir değer elde etmesi bunu destekler niteliktedir. Algoritmaların 30 koşmanın kaç tanesinde optimum küme sayısını bulduğuna bakılacak olursa, Gbest-ABC ve PSO Wisconsin ve Diabet data setlerinde hep 2 kümeyi bularak önemli bir başarı sağladığı Şekil 4'te görülmektedir. Gbest-ABC ayrıca Iris veri setinin kümelenmesinde 3 kümeyi hiç sapmadan bularak yine bu konuda önemli bir başarı ortaya koymuştur. Ancak bir diğer küresel en iyi tabanlı ABC algoritması olan GABC Iris veri setinde hiçbir zaman 3 kümeyi bulamamıştır. Diğer üç kümeli Wine data setinde ise Gbest-ABC ve PSO optimum küme elde etmede yine başarı sağladığı görülmüştür. Ancak, GABC ve ABC Wine data setinde optimum küme elde edilmesi konusunda yeterli başarıyı gösterememiştir. Vehicle, Dermatology ve Glass data setlerinde ise Gbest-ABC'nin yine optimum küme sayısı bulmada iyi performansı devam etmektedir. Buna karşın GABC'nin sadece Dermatology'de optimum küme sayısı bulmada başarılı olduğu, PSO'nun ise 3'ten fazla optimum küme sayısı bulmada performansının düştüğü görülmüştür. Sonuç olarak Gbest-ABC'nin otomatik data seti kümelemede de diğerlerinden iyi performans sağlamaktadır.

Tablo 3. Data Kümeleme İçin VI İndeks Sonuçları (The Results of VI Index for Data Clustering)

Datalar	Gbest ABC	GABC	ABC	PSO
Wisconsin	**0,0356** (0,0029)	0.1028 (0.0167)	0,1028 (0,0211)	0,0504 (0,0074)
Diabet	**0,0175** (0,0011)	0.1194 (0.0560)	0,1344 (0,0772)	0,0284 (0,0079)
Iris	**0,0871** (0,0082)	0.1283 (0.0141)	0,0881 (0,0074)	0,1269 (0,0329)
Wine	**0,1664** (0,0206)	0.2861 (0.0256)	0,2688 (0,0286)	0,3388 (0,0291)
Vehicle	**0,1316** (0,0316)	0.3397 (0.0403)	0,3379 (0,0516)	0,3986 (0,0704)
Dermatology	**0,3078** (0,0347)	0.4197 0.0395	0,4279 (0,0398)	0,4617 (0,1068)
Glass	**0,1075** (0,0354)	0.2221 (0.0527)	0,2368 (0,0778)	0,2269 (0,0717)

Tablo 4. Data Kümeleme İçin Ortalama Optimal Küme Sayıları (The Mean Number of Optimal Clusters for Data Clustering)

Datalar	No	Gbest ABC	GABC	ABC	PSO
Wisconsin	2	**2** (0)	2,0667 (0,254)	2,0667 (0,254)	**2** (0)
Diabet	2	**2** (0)	2,0333 (0,1825)	2,2 (0,5509)	**2** (0)
Iris	3	**3** (0)	4,1333 (0,3457)	3,1 (0,3051)	2,9667 (0,3198)
Wine	3	**3,1333** (0,3457)	3,6666 (0,8022)	3,8667 (0,8603)	**3,1333** (0,3457)
Vehicle	4	**4,2333** (0,4302)	4,8333 (0,8339)	5,1333 (1,1366)	3,7333 (0,7849)
Dermatology	6	5,8333 (0,5921)	**5,9333** (0.6914)	5,7333 (0,6397)	5,3 (0,5349)
Glass	6	5,4333 (0,5040)	5.4333 (0.6260)	**5,5333** (0,6288)	3,0333 (1,2452)



Şekil 4. Data kümeleme için 30 koşma üzerinden elde edilen toplam optimum küme sayısı (The total numbers of obtained optimum clusters for data clustering over 30 simulations)

7. SONUÇ (CONCLUSION)

Bu çalışmada, ABC algoritmasına vektörel çözüm üretme ve küresel en iyi bölgede çözüm arama mekanizmaları eklenerek ABC algoritmasının (Gbest-ABC) otomatik kümeleme problemine uyarlaması yapılmıştır. Gbest-ABC data ve görüntü kümeleme problemlerine uygulanmış, elde edilen sonuçlar çerçevesinde Gbest-ABC'nin GABC, ABC ve PSO'ya göre görüntü ve data kümelemedeki yetkinliği ve kabiliyeti net bir şekilde ortaya çıktığı ve otomatik kümeleme problemlerinde rahatlıkla kullanılabileceği neticesine varılmıştır.

KAYNAKLAR (REFERENCES)

1. Demiralay, M. **Hiyerarşik kümeleme metotları ile veri madenciliği uygulamaları**, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, 2005.
2. Jain, A.K., Murty, M.N., Flynn, P. J., "Data Clustering: A Review", **ACM Computing Surveys,** Cilt 31, Sayı 3, 264-323, 1999.
3. Manning, C.D., Schutze, H., **Foundations Of Statistical Natural Language Processing**, 1999.
4. Omran, M., **Particle Swarm Optimization Methods for Pattern Recognition and Image Processing**, Ph.D., Thesis, University of Pretoria, Environment and Information Technology, 2004.
5. Isık, M., **Bölünmeli kümeleme yöntemleri ile veri madenciliği uygulamaları**, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2006.
6. Yesilbudak, M., Kahraman, H. T., Karacan, H., "Object Oriented Agglomerative Hierarchical Clustering Model in Data Mining", **Journal of the Faculty of Engineering and Architecture of Gazi University,** Cilt 26, Sayı 1, 27-39, 2011.
7. Hamerly, G., Elkan, C., "Alternatives to the K-means Algorithm that Find Better Clusterings," **Proceedings of the ACM Conference on Information and Knowledge Management (CIKM-2002)**, 600–607, 2002.
8. MacQueen, J., "Some methods for classification and analysis of multivariate observations," **5th Berkeley Symp. Math. Stat. Probability**, 281–297, 1967.
9. Bezdek, J.C., **Pattern Recognition with Fuzzy Objective Function Algoritms**. Plenum Press, New York, 1981.
10. Durmuş, M.S., **Veri kümeleme algoritmalarının performansları üzerine karşılaştırmalı bir çalışma**, Yüksek Lisans Tezi, Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, 2005.
11. Omran, M. G. H., Salman, A., Engelbrecht, A. P. "Dynamic clustering using particle swarm optimization with application in image segmentation", **Pattern Analysis and Applications,** Cilt 8, Sayı 4, 332-344, 2006.
12. Das, S., Abraham, A., Konar, A., "Automatic Clustering Using an Improved Differential Evolution Algorithm", **IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,** Cilt 38, Sayı 1, 218-237, 2008.
13. Das, S., Konar, A., "Automatic image pixel clustering with an improved differential evolution", **Applied Soft Computing,** Cilt 9, Sayı 1, 226-236, 2009.
14. Maulik, U., Saha, I., "Automatic Fuzzy Clustering Using Modified Differential Evolution for Image Classification", **IEEE Transactions on Geoscience and Remote Sensing,** Cilt 48, Sayı 9, 3503-3510, 2010.
15. Ouadfel, S., Batouche, M., Taleb-Ahmed, A., "A Modified Particle Swarm Optimization Algorithm for Automatic Image Clustering", **International Symposium on Modelling and Implementation of Complex Systems, MISC'2010**, 49-57, 2010.
16. Kuo, R.J., Syu, Y.J., Chen, Z.Y., Tien, F.C., "Integration of particle swarm optimization and genetic algorithm for dynamic clustering", **Information Sciences,** Cilt 195, 124-140, 2012.
17. Bandyopadhyay, S., Saha, S., "**Point Symmetry-Based Distance Measures and Their**

Applications to Clustering," in Unsupervised Classification, ed: Springer Berlin Heidelberg, 2013, pp. 93-123.

18. Xie, X., Beni, G., "Validity measure for fuzzy clustering", **IEEE Trans. Pattern Anal. Machine Learning,** Cilt 13, Sayı 4, 841–846, 1991.

19. Bandyopadhyay, S., Saha, S., "A Point Symmetry-Based Clustering Technique for Automatic Evolution of Clusters", **IEEE Transactions on Knowledge and Data Engineering,** Cilt 20, Sayı 11, 1441-1457, 2008.

20. Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K., "A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA", **IEEE Transactions on Evolutionary Computation,** Cilt 12, Sayı 3, 269-283, 2008.

21. Saha, S., Bandyopadhyay, S., "A symmetry based multiobjective clustering technique for automatic evolution of clusters", **Pattern Recognition,** Cilt 43, Sayı 3, 738-751, 2010.

22. Saha, S., Bandyopadhyay, S.İ, "A generalized automatic clustering algorithm in a multiobjective framework", **Applied Soft Computing,** Cilt 13, Sayı 1, 89-108, 2013.

23. Karaboga, D., Basturk, B., "On the performance of artificial bee colony (ABC) algorithm", **Applied Soft Computing,** Cilt 8, Sayı 1, 687-697, 2008.

24. Karaboga, D., Okdem, S., Ozturk, C., "Cluster Based Wireless Sensor Network Routing using Artificial Bee Colony Algorithm", **Wireless Networks,** Cilt 18, Sayı 7, 847-860, 2012.

25. Eke, İ., Taplamacıoglu, M.C., Kocaarslan, İ., "Design of Robust Power System Stabilizer Based On Artifical Bee Colony Algorithm", **Journal of the Faculty of Engineering and Architecture of Gazi University,** Cilt 26, Sayı 3, 683-690, 2011.

26. Karaboga, D., Ozturk, C., "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", **Applied Soft Computing,** Cilt 11, Sayı 1, 652-657, 2011.

27. Hancer, E., Ozturk, C., Karaboga, D., "Artificial Bee Colony Based Image Clustering Method," **IEEE Congress on Evolutionary Computation (CEC2012)**, Brisbane, Australia, 1-5, 2012.

28. Ozturk, C., Hancer, E., Karaboga, D. "Improved Clustering Criterion for Image Clustering with Artificial Bee Colony Algorithm", **Pattern Analysis and Applications,** 2014.

29. Karaboga, D., **An idea based on honey bee swarm for numerical optimization**. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

30. Dorigo, M., **Optimization Learning And Natural Algorithms**, Ph.D. Thesis, Politecnico Di Milano, Italy, 1992.

31. Eberhart, R., Kennedy, J., "A new optimizer using particle swarm theory," **6th International Symposium on Micro Machine and Human Science**, 1995.

32. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M., **The bees algorithm**, Technical report. Cardiff University, UK, Manufacturing Engineering Centre, 2005.

33. Yang, X., **Engineering optimizations via nature-inspired virtual bee algorithms**, Springer, 2005

34. Lucic, P., Teodorovic, D., "Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence", **Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis**, Sao Miguel, Azores Islands, Portugal, 441–445, 2001.

35. Teodorovic, D., Dell'orco, M., "Bee colony optimization - a cooperative learning approach to complex transportation problems," **16th mini-EURO Conference on Advanced OR and AI Methods in Transportation**, 51–60, 2005.

36. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., "A comprehensive survey: artificial bee colony (ABC) algorithm and applications", **Artificial Intelligence Review,** Cilt 42, Sayı 1, 21-57, 2014.

37. Ozturk, C., **Yapay Sinir Ağlarının Yapay Arı Koloni Algoritması İle Eğitilmesi**, Doktora Tezi., Erciyes Universitesi, Fen Bilimleri Enstitüsü, 2011.

38. Dunn, J. C. "Well separated clusters and optimal fuzzy partitions", **J. Cybern.,** Cilt 4, 95–104, 1974.

39. Calinski, R. B., Harabasz, J. "A dendrite method for cluster analysis", **Commun. Stat.,** Cilt 3, No 1, 1–27, 1974.

40. Davies, D. L., Bouldin, D. W., "A cluster separation measure", **IEEE Trans. Pattern Anal. Mach. Intell.,** Cilt 1, No 1, 224–227, 1979.

41. Chou, C. H., Su, M. C., Lai, E., "A new cluster validity measure and its application to image compression", **Pattern Analysis and Applications,** Cilt 7, No 2, 205–220, 2004.

42. Turi, R. H., **Clustering-Based Colour Image Segmentation**, Ph.D. Thesis, Monash University, Australia, 2001.

43. Omran, M., Engelbrecht, A., Salman, A., "**Particle Swarm Optimization for Pattern Recognition and Image Processing**," in Swarm Intelligence and Data Mining. Cilt. 1, Abraham, A., Grosan, C.Ramos, V., Eds., ed: Springer-Verlag, SCI Series 'Studies in Computational Intelligence', 2006.

44. Jadhav, H.T., Sharma, U., Patel, J., Roy, R., "Gbest guided artificial bee colony algorithm for emission minimization incorporating wind power", **11th International Conference on Environment and Electrical Engineering (EEEIC2012)**, 1064-1069, 2012.